Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 17(3): 191-197, Sep. 2019 Regular paper

KNE: An Automatic Dictionary Expansion Method Using Use-cases for Morphological Analysis

Chung-Hyeon Nam and Kyung-Sik Jang*, *Member*, *KIICE*

Department of Computer Engineering, Korea University of Technology and Education, Cheonan 31253, Korea

Abstract

Morphological analysis is used for searching sentences and understanding context. As most morpheme analysis methods are based on predefined dictionaries, the problem of a target word not being registered in the given morpheme dictionary, the so-called unregistered word problem, can be a major cause of reduced performance. The current practical solution of such unregistered word problem is to add them by hand-write into the given dictionary. This method is a limitation that restricts the scalability and expandability of dictionaries. In order to overcome this limitation, we propose a novel method to automatically expand a dictionary by means of use-case analysis, which checks the validity of the unregistered word by exploring the use-cases through web crawling. The results show that the proposed method is a feasible one in terms of the accuracy of the validation process, the expandability of the dictionary and, after registration, the fast extraction time of morphemes.

Index Terms: Dictionary expansion, Natural language processing, Unknown word detection, Use-case analysis

I. INTRODUCTION

In natural language processing, morpheme analysis divides a sentence into a set of morphemes, which are the smallest units of speech that have their own meaning. Morphological analysis is used in various natural language processing fields as a method of computational linguistics, and is used in such applications as search engines and identification of context and intention [1]. In information retrieval, when a user inputs a query to a search engine, the retrieval speed and accuracy can be improved by filtering out unnecessary words or extracting meaningful nouns through morphological analysis. In a question-answer system, morphological analysis is used to determine the content of the conversation when a user attempts to communicate with a computer. Generally, morphological analysis processes are comprised of four steps: (1) generating a candidate according to the grammar rules, (2) checking the binding constraints between morphemes, (3) generating the candidate list of morphemes satisfying the given conditions and (4) selecting the most preferable morphemes from amongst the candidates based on their probability value. Most algorithms which generate candidate lists of morphemes, such as tabular parsing, Aho-Corasick, algorithms employing a hidden Markov model (HMM), Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN) are based on a predefined morpheme dictionary [2-4]. The words registered in this morpheme dictionary can be classified into two categories. One is a closed class, it is fixed and does not change or evolve over time. The other is an open class that can be enlarged over time [5]. The words belonging to the open class, such as nouns and verbs, do not change their own meaning, instead, new words such as coined or compound words are added over time.

On the other hand, words belonging to the closed class, such as particles, suffixes etc., are rarely altered or gener-

Received 08 August 2019, **Revised** 09 September 2019, **Accepted** 09 September 2019
***Corresponding Author** Kyung-Sik Jang (E-mail: ksjang@koreatech.ac.kr, Tel: +82-41-560-1352)
Department of Computer Engineering, Korea University of Technology and Education, Cheonan 31253, Korea.

Open Access https://doi.org/10.6109/jicce.2019.17.3.191 print ISSN: 2234-8255 online ISSN: 2234-8883

ated. The unregistered word problem, in which a target word is unregistered in the predefined dictionary, is one of the critical problems of morphological analysis. This problem occurs predominately with words from the open class rather than those of the closed class. Currently, a probabilistic interpretation method is used to solve the unregistered word problem. However, if the boundary between a noun and the words adjoining it is ambiguous, the probabilistic interpretation is unable to perform the extraction. For example, when the phrase "한기대" is entered, the morphological analysis predicts three cases: "한기대", "한기 + 대" and "한 + 기대". Actually, the word "한기대" is an abbreviation of the compound noun "한국기술교육대학교". However, if "한기대" is not already registered in the morpheme dictionary, it can be misjudged to consist of nouns such as the words "한기" and "대" or "한" and "기대".

As the number of newly created proper nouns, coined words and compound words increases due to the sociality of language, solving the unregistered word problem is becoming a critical task to overcome in enhancing the performance of morphological analysis. The current practical solution of such unregistered words is for people to manually add them into the given dictionary. The manual registration of words is a limitation that restricts the scalability and expandability of morpheme dictionaries. In order to overcome this limitation, we propose a novel method to automatically expand the morpheme dictionary by means of use-case analysis, which checks the validity of the unregistered word by exploring its use-cases through Internet searching.

Our proposed solution to the unregistered word problem focuses on the fact that when people come across unknown words, they tend to search the Internet or consult a dictionary to understand its meaning. If a word that does not exist in the morpheme dictionary is found in the process of morphological analysis, the problem is solved using Web crawling to search the Internet for unregistered words, similar to the behavior of a person searching the dictionary or Internet for an unknown word. Specifically, the validity of the unregistered word is checked by analyzing the web pages obtained by inputting the unregistered word into the Internet search engine, and examining how often the unregistered word is used within those pages.

Using the proposed method, we can overcome the inconvenience of manually expanding the morpheme dictionary with previously unregistered words. Furthermore, we can improve the performance of morphological analysis by automatically registering new words in the morpheme dictionary.

II. RELATED WORKS

The morpheme analyzer uses analysis algorithms to select the correct tags for each word. If there are words in the input sentence that are not registered in the dictionary, the word is divided into smaller units. If these separated units are in the dictionary, an error occurs in attaching a different tag. For this reason, unregistered words in the tagging process of morpheme analysis cause fatal performance degradation.

The existing methods dealing with unregistered words are largely classified as either a method of estimating the boundary of a word using the sentence structure, a method of estimating using language characteristics, or a method of probabilistic estimation using machine learning.

First, there are two methods of estimating the boundary of a word using the sentence structure: a method using SVM (Support Vector Machine) and a method using POS (Part-Of-Speech) Tagger, these estimate the boundary between an unregistered word and a registered word by analyzing sentence structure [6-8]. These methods can accurately estimate and extract the boundary of a word when there are few unregistered words in the input sentence. However, the possibility of error increases when there are many unregistered words, as the structure of the sentence cannot be so effectively analyzed.

Second, in the process of attaching morpheme tags to each word of the input sentence, the morpheme of the word is estimated using language characteristics. A tag is attached to the words most frequently appearing of the input sentence words, then the remaining words are extracted [9]. With this method, similarly to the method using the sentence structure, correct tags can be attached when there are few unregistered words, because the surrounding words are also tagged. However, when there are many unregistered words, the possibility of error increases.

Lastly, among the machine learning algorithms, there is SVM, which is a supervised learning method, Decision Tree, which is an unsupervised learning method, and deep learning. Tags of the words adjoining frequently appearing words are used as part of a probabilistic approach to predict what tags are likely. However, this approach cannot create learning data, because it cannot use a probabilistic approach to learn about all types of sentences, and cannot cope with sentence structures that have not been previously learnt [10-12].

These limitations cannot be avoided since the aforementioned methods operate by estimating the morphology of a word in various forms. In this paper, we assume that when improving the accuracy of morpheme analysis, it is more effective to extend the morpheme dictionary rather than to estimate. Therefore, we propose a method of automatic dictionary expansion.

III. AUTOMATIC DICTIONARY EXPANSION

In this paper, we propose a Korean Noun-dictionary

Expander (KNE) that applies an automatic morphological dictionary expansion algorithm to a noun dictionary. The KNE, as shown in Fig. 1, consists of the processes of querying a noun and non-noun, dividing an input word character by character, querying a preconstructed noun and non-noun dictionary, creating a registered candidate noun and finally verifying that the noun actually exists in practice.

A. Dictionary Generation

The dictionary employed for noun extraction in KNE consists of two smaller dictionaries, a noun and a non-noun dictionary. The noun dictionary is a noun-only dictionary that extracts only the nouns of NIADic [13], the Hangul morpheme dictionary of the K-ICT Big Data Center. The non-noun dictionary extracts the text by downloading a Wikipedia dump file. After extracting the words based on spacing, it checks whether each noun is included in the noun dictionary. If a noun is included in a word, the remaining part of the word is defined as non-noun and stored alongside the frequency of its occurrence.

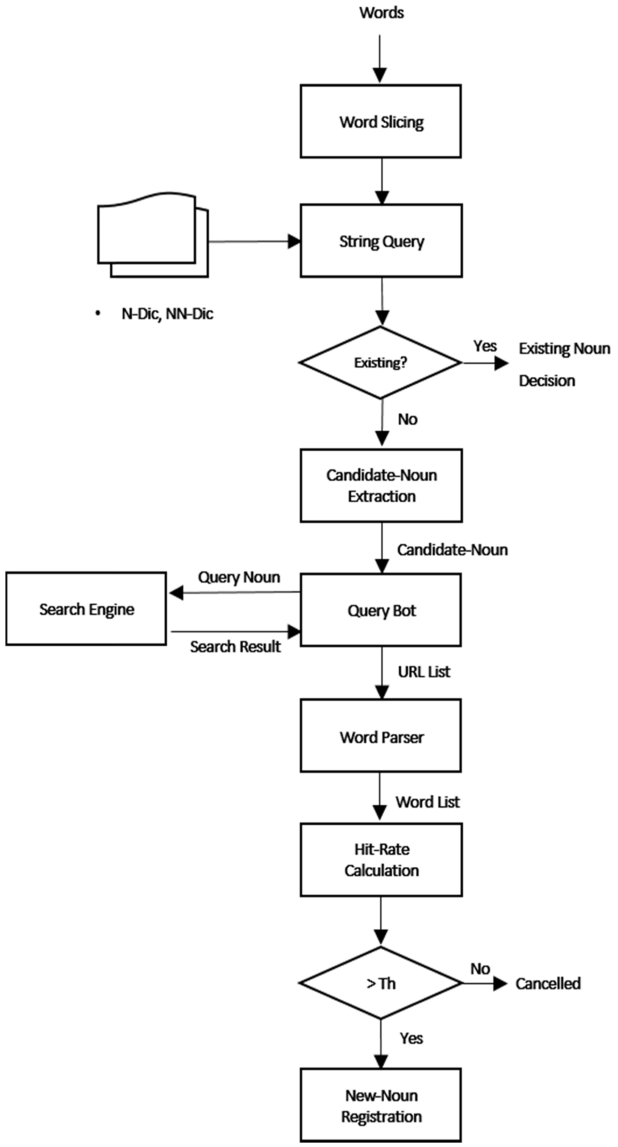

Fig. 1. Flow diagram of KNE operation.

B. Word Slicing

Word slicing is performed by slicing the input word from right to left, to generate an L+R structure set. In Korean, a word is in a L+R structure, and the words that embellish nouns are followed by nouns. Here, L is the noun part and R is the non-noun part. Reflecting these Korean characteristics, when a word is entered, the KNE creates a set of [L-String, R-String] structures by moving one character at a time from the L to R string, to make a noun and non-noun candidate. For example, when the phrase "한기대를" is entered, the L+R structure set becomes [("한기대" + "를"), ("한기" + "대를"), ("한" + "기대를")].

C. String Query

String query is the process of examining whether a string generated through word slicing is registered in a preconstructed noun or non-noun dictionary. For each element of the [L-String, R-String] set created in the word slicing process, the L-String is checked using the noun dictionary and the R-String is checked using the non-noun dictionary. If an L-/R-String is registered in both a noun/non-noun dictionary, L-String is determined as a noun. Otherwise, it will be judged as a non-registered noun and the process moves on to the next step to decide whether to register a new entity or not.

D. Candidate-Noun Extraction

All the R-Strings are extracted from the [L-String, R-String] set generated by word slicing for the registered candidate words, and are compared with the non-noun dictionary. The R-String with the highest frequency is selected as the non-noun part of the input word. The remaining part of the input word after the R-String has been removed (that is, the L-String) is determined as the registration candidate noun. For example, if the input phrase is "한기대를", the extracted R-Strings are ["를", "대를", "기대를"]. If the frequency values of the R-Strings stored in the non-noun dictionary are {"를" : 3789, "대를" : 150, "기대를" : 0}, "를" , having the highest frequency, is selected as the non-noun part and "한기대" becomes a candidate for registration after "를" has been removed from the input word.

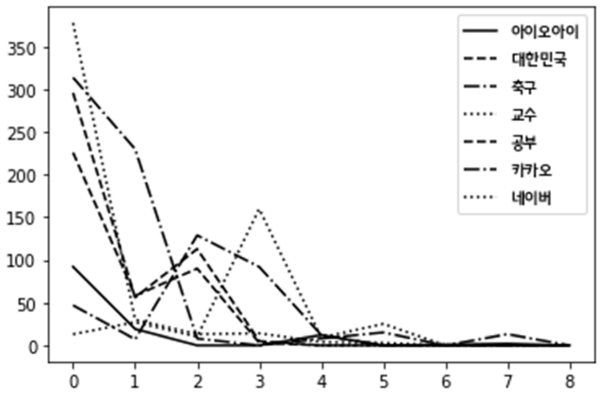

Fig. 2. Number of words included in the top 9 resulting web pages of a search engine.

E. Query Bot

The query bot queries the search engine for the registration candidate-noun, to collect the URLs for the web pages that come up as search results. When querying a word, most search engines prioritize web pages (URLs) that are most relevant to a word, or have a high frequency of word exposure. Therefore, it is assumed that the web pages listed at the top of the search results, indicated by the URLs, contain the most information about the registered candidate-noun. Fig. 2 shows the frequency of exposure of a searched word on the top 9 web pages returned when a specific word is searched for on a popular search engine.

As can be seen in Fig. 2, the web page at the top of the search results contained the searched word most frequently, and the frequency of exposure decreased further down the list. Based on these measurement results, the query bot is set to collect only the top 5 URLs of the search results.

F. Word Parser

Word parsing is a process of collecting words containing the registered candidate-noun in a web page, it is shown in Fig. 3. First, all the web pages corresponding to the URL list collected from the query bot are analyzed using the web crawler. Then, only the pure texts are extracted from the web page (excluding the HTML tags and java script grammar) and are made into a list of words separated by whitespace. The words containing the registered candidate-noun are extracted from the word list and the target word list is created. The L-String of the L+R structure of the words included in the target word list become candidates for registration.

G. Hit-rate Calculation

Hit-rate calculation is a process of judging whether to select a definite noun by analyzing a word list containing the

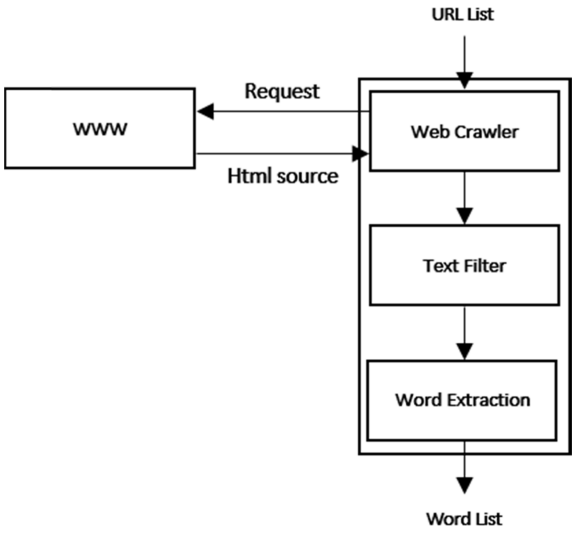

Fig. 3. Architecture of word parser.

registered candidate-noun. It is determined by calculating the hit rate for the R-String portion of the L+R structure of the target word. The hit rate is defined as follows and has a value between 0 and 1.

$$\text{Hit Rate} = \frac{H}{S}. \tag{1}$$

Here S is the total number of words included in the word list, and H is the number of words of which the R-String is registered in the non-noun dictionary. In fact, the R-String part of the word may be nothing, or it may not be the suffixes or short words that immediately follows a noun or pronoun (e.g. a postpositional particle or ending) so the hit rate is calculated to determine whether to approve the registered candidate-noun or not. If the hit rate of the registered candidate-noun exceeds the threshold value (Vth), it becomes a definite noun and is added to the noun dictionary. Otherwise, the noun registration is cancelled. Our threshold value (Vth) is determined by experimental results. The average hit rate was 0.81 for 534 coined words found in Wikipedia. Therefore, in KNE, the hit-rate threshold was set to 0.9 for successful noun registration.

IV. EXPERIMENTAL RESULTS

While conventional morpheme analyzers have used a static dictionary, we have proposed a dynamic dictionary expansion method that can automatically add new morphemes to the existing dictionary without stopping the morphology analysis process. The procedure of dynamic dictionary expansion consists of 3 steps : (1) definition of the candidate set, (2) selection of the most preferable candidate, and (3) validation of the candidate word. First, the definition of candi-

date set proceeds by slicing the unknown word from right to left, character by character. Then, the most preferable candidate is selected by comparing the frequency with which the R-Strings of the sliced words occur in the non-noun dictionary, which includes the particles and endings used in normal sentences. Finally, the correctness of the selected candidate is verified by performing use-case analysis through web page searching on the Internet.

In order to prove the effectiveness of the automatic noun dictionary expansion method presented in this paper, we experimented upon proper nouns (personal names, region names, organizations) and coined words, which are often included as unregistered nouns.

As shown in Table 1, the Name dataset used in the experiment with 10,700 randomly combined Korean names and particles, and the Location and Organization datasets had 10561 and 14889 entries respectively, using data from the Korean gazette [14] provided by the National Institute of the Korean Language. In addition, 434 coined words from Wikipedia were used as subjects to test the performance of the method to classify new words.

Data length affects the noun extraction performance. Table 2 shows the mean length, minimum length, maximum length and variance of the datasets used in the experiment. Since the Name dataset is a set of randomly combined postpositional particles in Korean names, it has an average length of 4.8 words, a minimum and maximum length of 4 and 6 words respectively, and a variance value of 0.56. Location and Organization dataset entries have an average length value of 5.07, similar to the Name dataset, but the variance values were 5.37 and 5.36 respectively, indicating that the data length is not as evenly distributed as it is in the Name dataset. In addition, the Coined Word dataset has a lower mean value than the Name dataset, but the variance value is 0.64, indicating that the data length has a wide distribution.

First, to compare the execution time, the data set was processes by the KNE presented in this paper and compared to the well-known Korean morphological analyzers KKMA, Komoran, Hannanum, and OKT. Table 3 shows the average execution time of each extractor for the data set presented in the experiment. KNE took a long time with an average of 1280 ms using only a basic noun dictionary (unexpanded-case). This is because the time required for approving the unregistered word (web crawler search time) is included. With the automatically expanded-case, noun extraction was possible at a very high speed (0.1 ms average) compared to other analyzers. In conclusion, when KNE discovers an unregistered word, the initial step of determining whether to register an unregistered word or not and then automatically registering it takes a relatively long time. After registration, results show that nouns were extracted at a high speed.

Table 4 shows the results found by measuring the average execution speed of each extractor, and the accuracy of noun extraction for the human name, region name and organization name datasets. In this table, KNE assumes that the dictionary is expanded-case. If the word is short, such as in Name, KNE's performance was better than other extractors. However, in the case of data such as Location and Organization, with long words and compound nouns, performance was similar to that of other extractors. Of the other extractors, Hannanum showed the highest performance.

Table 5 shows the extraction performance for 434 coined words. KNE correctly analyzed 360 coined words, Hannanum was the most accurate extractor, and the accuracy of KKMA was the lowest with 263.

Finally, the extraction performance was compared when two extractors were used in parallel to mutually complement each other's extractive performance. As shown in Table 6, the performance was better when the KNE and other extractors were used together than when the extractors were

Table 1. Dictionary data source and number

Category	# of words	Base
Name	10,700	Web
Location	10561	Korean gazette
Organization	14889	
Coined Word	434	Wikipedia

Table 2. Dataset statistical properties

Category	Mean Length	Min Length	Max Length	Variance
Name	4.80	4	6	0.56
Location	5.07	1	28	5.37
Organization	4.77	1	28	5.36
Coined Word	2.57	1	7	0.64

Table 3. Time comparison of noun extractor KNE and other extractors

Extractor		Time (ms)
KNE	Unexpanded-case	1280
	Expanded-case	0.1
KKMA		27
Komoran		3
Hannanum		1
OKT		2

Table 4. Performance and time spent by each extractor

Extractor	Name		Location		Organization	
	Acc	Time	Acc	Time	Acc	Time
KNE	99.9	0.1	36.4	0.1	42.4	0.1
KKMA	67.2	27	22.0	23	26.0	14
Komoran	27.7	0.3	38.0	1	44.1	2
Hannanum	93.0	0.9	79.8	1	81.1	1
Okt	74.7	1	38.7	6	35.0	0.8

Table 5. Accuracy of KNE and other extractors

Extractor	Nouns
KNE	360
KKMA	263
Komoran	252
Hannanum	373
OKT	305

Table 6. Accuracy of combination of both KNE and other extractors

Extractor	Nouns
KKMA + KNE	407
Komoran + KNE	405
Hannanum + KNE	417
OKT + KNE	406

used separately. When Hannanum and KNE were combined, 417 words out of 434 coined words could be extracted. Through these experiments, we confirmed that the scalability of the dynamic dictionary is satisfied and that the correctness of newly added words is well validated, by checking that a performance improvement is achieved for morphological analyzers with a dynamic dictionary.

V. CONCLUSION

In general, the performance of the morpheme analyzer depends heavily on the size of the dictionary used in the analysis. While there are no problems in analyzing pre-registered words, various methods of processing unregistered words have been proposed. In this paper, we proposed an automatic Korean Noun-dictionary Expander (KNE) using Internet searching as an efficient method to process unregistered words in morpheme analysis. In KNE, when the unregistered word is found, the possibility of noun registration is checked by analyzing the use-cases of the unregistered word, using a search engine and web crawler to do so. In this way, the performance of the morphological extractor can be improved by automatically expanding the size of its dictionary. As a result of our experiments using proper nouns (10,700 names, 14,889 organizations, 10,561 locations) and an unregistered noun dataset, KNE shows a very fast execution speed after one registration time, and a relatively high accuracy of extracted nouns. In addition, it was confirmed that performance can be complemented by using KNE alongside other extractors. We applied KNE only to noun extraction in this paper, in the future, we anticipate that it can be successfully applied to other types of morpheme as well. Currently, the method is applied only to the extraction of unregistered nouns. However, for the extraction of other morphemes, the sentence structure and the relation analysis between words should be further processed to improve the accuracy of the extraction.

ACKNOWLEDGMENTS

This paper was supported by the Education and Research Promotion Program of KOREATECH in 2017.

REFERENCES

[1] L. Marquez, L. Padro and H. Rodriquez, "A machine learning approach to PoS tagging," *Machine Learning*, vol. 39, no. 1, pp. 59-91, 2000. DOI: 10.1023/A:1007673816718.

[2] A. V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," *Communications of the ACM*, vol. 18, no. 6, pp. 333-340, 1975. DOI: 10.1145/360825.360855.

[3] J. V. Gael, A. Vlachos and Z. Ghahramani, "The infinite HMM for unsupervised PoS tagging," in *Proceeding of 2009 Conference on Empirical Methods in Natural Langue Processing*, Singapore, vol. 2, pp. 678-687, 2009.

[4] C. N. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging." in *Proceeding of the 31st International Conference on Machine Learning*, Beijing, China, vol. 32, pp. 1818-1826, 2014.

[5] A. Y. Aikhenvald, "The Art of Grammar: A Practical Guide." Oxford University Press; UK ed. Edition, pp. 99, 2015.

[6] R. Hiraoka, H. Tanaka, S. Sakti, G. Neubing and S. Nakamura, "Personalized unknown word detection in non-native language reading using eye gaze." in *Proceeding of the 18th ACM International Conference on Multimodal Interaction*, pp. 66-70, 2016. DOI: 10.1145/2993148.2993167.

[7] A. Mikheev, "Automatic rule induction for unknown-word guessing." *Computational Linguistic*, vol. 23, pp. 405-423, 1997.

[8] K. Erk, "Unknown word sense detection as outlier detection," in *Proceeding of the Human Language Technology Conference of the North American Chapter of the ACL*, pp. 128-135, 2006. DOI: 10.3115/1220835.1220852.

[9] W. Pang, X. Fan, Y. Gu and J. Yu, "Chinese unknown words extraction based on word-level characteristics." in *Proceeding of the Ninth International Conference on Hybrid Intelligent Systems*, vol. 39, pp. 361-366, 2009. DOI: 10.1109/HIS.2009.77.

[10] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou and Y. Bengio, "Pointing the unknown words," in *Proceeding of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 140-149, 2016. DOI: 10.18653/v1/P16-1014.

[11] T. Nakagawa, T. Kudoh and Y. Matsumoto, "Unknown word guessing and part-of-speech tagging using support vector machines," in *Proceeding of the Sixth Natural Language Processing Pacific Rim Symposium*, 2001.

[12] G. S. Orphanos and D. N. Christodoulakis, "POS Disambiguation and unknown word guessing with decision trees," in *Proceeding of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, pp. 134-141, 1999. DOI: 10.3115/977035.977054.

[13] K-ICT Big Data Center [Internet], Available: https://kbig.kr/portal/kbin/knowledge/files/bigdata_report.page?bltnNo=10000000016451.

[14] National Institute of Korean Language and Information Sharing Center [Internet], Available: https://ithub.korean.go.kr/user/main.do.

Chung-Hyeon Nam
received a B.S degree in computer engineering from the Korea University of Technology and Education in 2019. His current interests include natural language processing and web crawling.

Kyung-Sik Jang
received a B.S degree in electronic engineering from Korea University, 1987, an M.S. degree in electrical and electronic engineering from KAIST in 1989, and a Ph.D in electrical and electronic engineering from the Tokyo Institute of Technology in 1998. His research interests include embedded systems and natural language processing.